Scrutinizing and Improving Impossible Differential Attacks: Applications to CLEFIA, Camellia, LBlock and Simon

Christina Boura, María Naya-Plasencia & Valentin Suder

ASIACRYPT 2014, Kaohsiung, Taiwan

December 8th, 2014
Impossible Differential Cryptanalysis

Impossible Differential Cryptanalysis:

- was introduced by Knudsen in 1998, and Biham, Biryukov & Shamir in 1999;
- is part of the Differential Cryptanalysis family...
- ...but uses a distinguisher of probability 0;
- is very efficient against iterated block ciphers.

L. R. Knudsen,
DEAL – A 128-bit cipher,
1998.

E. Biham, A. Biryukov and A. Shamir,
Cryptanalysis of Skipjack Reduced to 31 Rounds Using Impossible Differentials,
EUROCRYPT’99.
Impossible Differential Cryptanalysis: Scenario

- place an impossible differential \((\delta_X, \delta_Y)\) on \(r_\delta\) rounds;
- extend it by differentials \((\delta_{in} \rightarrow \delta_X)\) and \((\delta_{out} \rightarrow \delta_Y)\);
- evaluate the parameters:

  \[
  r_{in}, r_{out} : \text{number of rounds} \\
  c_{in}, c_{out} : \log \text{of the probabilities} \\
  k_{in}, k_{out} : \text{involved subkeys} \\
  |k_{in} \cup k_{out}| : \text{key entropy}
  \]
Finding an Impossible Differential

\[ \delta_X \]

\[ \delta_Y \]

- Miss-in-the-middle technique [BBS99];

- \( \mathcal{U} \)-method [Kim et al. 03];

J. Kim and S. Hong and J. Sung and C. Lee and S. Lee,
Impossible Differential Cryptanalysis for Block Cipher Structures,
INDOCRYPT’03.
Early-Abort Technique

J. Lu, J. Kim, N. Keller and O. Dunkelman,
Improving the Efficiency of Impossible Differential Cryptanalysis of Reduced Camellia and MISTY1,
CT-RSA’08.
Early-Abort Technique

J. Lu, J. Kim, N. Keller and O. Dunkelman,
Improving the Efficiency of Impossible Differential Cryptanalysis of Reduced Camellia and MISTY1,
CT-RSA’08.

'mustguess' keybits = \[
\begin{array}{c c c c}
- & - & - & \cdots
\end{array}
\]

Pairs w/ $\delta_{in}/\delta_{out}$
Early-Abort Technique

J. Lu, J. Kim, N. Keller and O. Dunkelman,

Improving the Efficiency of Impossible Differential Cryptanalysis of Reduced Camellia and MISTY1,

CT-RSA’08.

'mustguess' keybits =

Pairs w/ $\delta_{in}/\delta_{out}$
Early-Abort Technique

J. Lu, J. Kim, N. Keller and O. Dunkelman,
Improving the Efficiency of Impossible Differential Cryptanalysis of Reduced Camellia and MISTY1,
CT-RSA’08.

'mustguess' keybits =
## Existing Flaws

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Ref.</th>
<th>Type</th>
<th>Gravity</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLEFIA-128</td>
<td>[ZH08]</td>
<td>data</td>
<td>✗</td>
</tr>
<tr>
<td>CLEFIA-128</td>
<td>[T10]</td>
<td>unverifiable</td>
<td>-</td>
</tr>
<tr>
<td>Camellia</td>
<td>[WZF07]</td>
<td>big flaw</td>
<td>✗</td>
</tr>
<tr>
<td>Camellia-128</td>
<td>[WZZ08]</td>
<td>big flaw</td>
<td>✗</td>
</tr>
<tr>
<td>Camellia</td>
<td>[LKKD08]</td>
<td>small flaws</td>
<td>✓</td>
</tr>
<tr>
<td>LBlock</td>
<td>[MN1208]</td>
<td>small flaw</td>
<td>✓</td>
</tr>
<tr>
<td>Simon</td>
<td>[ALLW13,14]</td>
<td>big flaw</td>
<td>✗</td>
</tr>
<tr>
<td>Simon</td>
<td>[AL13]</td>
<td>data</td>
<td>✗</td>
</tr>
</tbody>
</table>
Objectives

- **Formalize** the evaluation of the complexities;
- **Automate** the whole process;
Objectives

- **Formalize** the evaluation of the complexities;
- **Automate** the whole process;

Results

- **Optimization** of previous attacks;
- **Development** of new techniques;
- **Application** to block ciphers (CLEFIA, Camellia, LBlock, **S**IMON) ⇒ **Best** Cryptanalysis.
Amount of Memory needed

\[ \frac{1}{2^{c_{in} + c_{out}}} \]
Amount of Memory needed

\[
\left( 1 - \frac{1}{2^{c_{in} + c_{out}}} \right)
\]
Amount of Memory needed

\[ P = \left(1 - \frac{1}{2^{c_{in}+c_{out}}}\right)^N < \frac{1}{2^{|k_{in} \cup k_{out}|}} \]
Amount of Memory needed

\[ P = \left(1 - \frac{1}{2^{c_{in} + c_{out}}} \right)^N < \frac{1}{2^{\left| k_{in} \cup k_{out} \right|}} \]
Amount of Memory needed

\[ P = \left(1 - \frac{1}{2^{c_{in} + c_{out}}} \right)^N < \frac{1}{2} \]

Since \( P \sim e^{-N(2^{-(c_{in} + c_{out})})} \), we will consider that \( N_{\text{min}} = 2^{c_{in} + c_{out}} \).
Amount of Memory needed

\[ P = \left( 1 - \frac{1}{2^{c_{in} + c_{out}}} \right) N < \frac{1}{2} \]

Since \( P \approx e^{-N(2^{-(c_{in}+c_{out})})} \), we will consider that \( N_{\text{min}} = 2^{c_{in}+c_{out}} \).

**Memory Complexity:** \( \min \left\{ N, 2^{|k_{in} \cup k_{out}|} \right\} \).
Amount of Data needed

To build these \( N \) pairs, we need \( C_N < 2^s \) plaintexts.
Amount of Data needed

To build these $N$ pairs, we need $C_N < 2^s$ plaintexts.

Data Complexity: $C_N$.

\[
C_N = \max \left\{ \min_{\delta \in \{\delta_{in}, \delta_{out}\}} \left\{ \sqrt{N2^s+1-|\delta|}, N2^s+1-|\delta_{in}|-|\delta_{out}| \right\} \right\} < 2^s.
\]
Time Complexity

\[ T_{\text{comp}} = C_N C_E + \]

- **Encrypt** all the data;
Time Complexity

\[ T_{\text{comp}} = C_N C_E + \left( 2|k_{\text{in}} \cup k_{\text{out}}| \frac{N}{2c_{\text{in}} + c_{\text{out}}} \right) C'_E \]

- Encrypt all the data;
- Early-Abort Technique
Evaluation of Complexities

Time Complexity

\[ T_{\text{comp}} = C_N C_E + \left( 2|k_{\text{in}} \cup k_{\text{out}}| \frac{N}{2c_{\text{in}} + c_{\text{out}}} \right) C'_E \]

- **Encrypt** all the data;
- **Early-Abort Technique**
  - Check **each** key step by step;
Time Complexity

\[ T_{\text{comp}} = C_N C_E + \left( \frac{2|k_{in} \cup k_{out}|}{2c_{in} + c_{out}} \right) C_E' \]

- **Encrypt** all the data;

- **Early-Abort Technique**
  - Check **each** key step by step;
  - Decrease the **number of pairs** in the list;
Time Complexity

\[ T_{\text{comp}} = C_N C_E + \left( 2|k_{\text{in}} \cup k_{\text{out}}| \frac{N}{2c_{\text{in}} + c_{\text{out}}} \right) C'_E + \frac{2|K|}{2|k_{\text{in}} \cup k_{\text{out}}|} \mathcal{P} 2|k_{\text{in}} \cup k_{\text{out}}| C_E. \]

- **Encrypt** all the data;

- **Early-Abort Technique**
  - Check each key step by step;
  - Decrease the number of pairs in the list;

- Test every key remaining in the candidate key set
Time Complexity

\[
T_{comp} = C_N C_E + \left(2|k_{in \cup k_{out}}| \frac{N}{2c_{in} + c_{out}} \right) C'_E + \frac{2|K|}{2|k_{in \cup k_{out}}|} P 2|k_{in \cup k_{out}}| C_E.
\]

- **Encrypt** all the data;

- **Early-Abort Technique**
  - Check each key step by step;
  - Decrease the number of pairs in the list;

- Test every key remaining in the candidate key set
  \[
  T_{comp} < 2|K| C_E.
  \]
Uniformized Formulas

\[ T_{\text{comp}} = C_N C_E + \left( 2|k_{\text{in}} \cup k_{\text{out}}| \frac{N}{2c_{\text{in}} + c_{\text{out}}} \right) C'_E + \mathcal{P} 2|K| C_E. \]

⇒ easy to use formulas;

⇒ more trade-offs;

⇒ automatic tool & systematic search;

⇒ development of new techniques;
New Techniques

- Multiple Impossible Differentials

- *State-Test Technique*
Example of an Application to CLEFIA

- **block size:**
  
  \[ 4 \times 32 = 128 \text{ bits} \]

- **key size:**
  
  128, 192, 256 bits

- **# of rounds:**
  
  18, 22, 26

\[ \begin{array}{c}
  P_{0}^{i-1} \quad P_{1}^{i-1} \quad P_{2}^{i-1} \quad P_{3}^{i-1} \\
  RK_{2i-2} \quad F_{0} \quad F_{1} \quad RK_{2i-1}
\end{array} \]

T. Shirai, K. Shibutani, T. Akishita, S. Moriai and T. Iwata,

The 128-Bit Blockcipher CLEFIA (Extended Abstract),

FSE’07.
Multiple Impossible Differentials

Formalize the idea of [Tsunoo et al. 08]

\[
\begin{array}{c|c|c}
\delta_X & \rightarrow & \delta_Y \\
\hline
(0, 0, 0, A) & (0, 0, 0, B) & (0, A, 0, 0) & (0, B, 0, 0) \\
\end{array}
\]

A | B
---|---
(0, 0, 0, α) | (0, 0, β, 0) | (0, β, 0, 0) | (β, 0, 0, 0)
(0, 0, α, 0) | (0, 0, 0, β) | (0, β, 0, 0) | (β, 0, 0, 0)
(0, α, 0, 0) | (0, 0, 0, β) | (0, β, 0, 0) | (β, 0, 0, 0)
(α, 0, 0, 0) | (0, 0, 0, β) | (0, β, 0, 0) | (0, β, 0, 0)

\[CN = 2^{113}\]
Multiple Impossible Differentials

Formalize the idea of [Tsunoo et al. 08]

\[
\begin{array}{c|c|c|c}
\delta X & \delta Y \\
\hline
(0,0,0,A) & (0,0,0,B) \\
(0,A,0,0) & (0,B,0,0) \\
\end{array}
\]

\[
\begin{align*}
A & \quad | \quad B \\
(0,0,0,\alpha) & | (0,0,\beta,0) & (0,\beta,0,0) & (\beta,0,0,0) \\
(0,0,\alpha,0) & | (0,0,0,\beta) & (0,\beta,0,0) & (\beta,0,0,0) \\
(0,\alpha,0,0) & | (0,0,0,\beta) & (0,0,\beta,0) & (\beta,0,0,0) \\
(\alpha,0,0,0) & | (0,0,0,\beta) & (0,0,\beta,0) & (0,\beta,0,0) \\
\end{align*}
\]

\[C_N = 2^{113} \implies C_N = 2^{113 - \log_2(24)}\]
State-Test Technique

Decrease the number of key bits to guess
State-Test Technique

Decrease the number of key bits to guess
**State-Test Technique**

Decrease the number of key bits to guess

![Diagram of State-Test Technique with cste, $RK_0$, $RK_1$, $RK_2$, $RK_3$, $F_0$, $F_1$, and $\delta_{in}$ and $\delta_X$.]
State-Test Technique

Decrease the number of key bits to guess

\[ B' = B \oplus \text{cste} = S_0(\oplus \text{cste}) \]
State-Test Technique

Decrease the number of key bits to guess

\[ B' = B \oplus \text{cste} = \text{cste} \oplus S_0(\text{cste} \oplus \text{cste}). \]

\[ |k_{in} \cup k_{out}| = 122 \text{ bits} \quad \Rightarrow \quad |k_{in} \cup k_{out}| = 122 - 16 + 8 \text{ bits} \]
## Comparison

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Rounds</th>
<th>Time</th>
<th>Data</th>
<th>Memory</th>
<th>Ref.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLEFIA-128</td>
<td>13/18</td>
<td>121.2</td>
<td>117.8</td>
<td>86.8</td>
<td>[MDS11]</td>
</tr>
<tr>
<td>state-test</td>
<td>13/18</td>
<td>116.90</td>
<td>116.33</td>
<td>83.33</td>
<td></td>
</tr>
<tr>
<td>multiple</td>
<td>13/18</td>
<td>122.26</td>
<td>111.02</td>
<td>82.60</td>
<td></td>
</tr>
<tr>
<td>multiple &amp; state-test</td>
<td>13/18</td>
<td>116.16</td>
<td>115.38</td>
<td>83.16</td>
<td></td>
</tr>
</tbody>
</table>
## Camellia

128-bit block

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Rounds</th>
<th>Time</th>
<th>Data</th>
<th>Memory</th>
<th>Ref.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Camellia-128</td>
<td>11/18</td>
<td>122</td>
<td>122</td>
<td>98</td>
<td>[LLGWLC12]</td>
</tr>
<tr>
<td></td>
<td>11/18</td>
<td><strong>118.43</strong></td>
<td><strong>118.4</strong></td>
<td><strong>92.4</strong></td>
<td></td>
</tr>
<tr>
<td>Camellia-192</td>
<td>12/24</td>
<td>187.2</td>
<td>123</td>
<td>155.41</td>
<td>[LLGWLC12]</td>
</tr>
<tr>
<td></td>
<td>12/24</td>
<td><strong>161.06</strong></td>
<td><strong>119.70</strong></td>
<td><strong>150.70</strong></td>
<td></td>
</tr>
<tr>
<td>Camellia-256</td>
<td>13/24</td>
<td>251.1</td>
<td>123</td>
<td>203</td>
<td>[LLGWLC12]</td>
</tr>
<tr>
<td></td>
<td>13/24</td>
<td><strong>225.06</strong></td>
<td><strong>119.71</strong></td>
<td><strong>198.71</strong></td>
<td></td>
</tr>
<tr>
<td>Camellia-256 †</td>
<td>14/24</td>
<td>250.5</td>
<td>120</td>
<td>125</td>
<td>[LLGWLC12]</td>
</tr>
<tr>
<td>state-test</td>
<td>14/24</td>
<td><strong>220</strong></td>
<td><strong>118</strong></td>
<td>173</td>
<td></td>
</tr>
</tbody>
</table>
**LBlock**

64-bit block, 80-bit key

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Rounds</th>
<th>Time</th>
<th>Data</th>
<th>Memory</th>
<th>Ref.</th>
</tr>
</thead>
<tbody>
<tr>
<td>LBlock</td>
<td>22/32</td>
<td>79.28</td>
<td>58</td>
<td>72.67</td>
<td>[KDH12]</td>
</tr>
<tr>
<td></td>
<td>23/32</td>
<td>74.06</td>
<td>59.6</td>
<td>74.6</td>
<td></td>
</tr>
</tbody>
</table>
### Results

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Rounds</th>
<th>Time</th>
<th>Data</th>
<th>Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simon-32/64</td>
<td>19/32</td>
<td>62.56</td>
<td>32</td>
<td>44</td>
</tr>
<tr>
<td>Simon-48/72</td>
<td>20/36</td>
<td>70.69</td>
<td>48</td>
<td>58</td>
</tr>
<tr>
<td>Simon-48/96</td>
<td>21/36</td>
<td>94.73</td>
<td>48</td>
<td>70</td>
</tr>
<tr>
<td>Simon-64/96</td>
<td>21/42</td>
<td>94.56</td>
<td>64</td>
<td>60</td>
</tr>
<tr>
<td>Simon-64/128</td>
<td>22/44</td>
<td>126.56</td>
<td>64</td>
<td>75</td>
</tr>
<tr>
<td>Simon-96/96</td>
<td>24/52</td>
<td>94.62</td>
<td>94</td>
<td>61</td>
</tr>
<tr>
<td>Simon-96/144</td>
<td>25/54</td>
<td>142.59</td>
<td>96</td>
<td>77</td>
</tr>
<tr>
<td>Simon-128/128</td>
<td>27/68</td>
<td>126.6</td>
<td>126</td>
<td>61</td>
</tr>
<tr>
<td>Simon-128/192</td>
<td>28/69</td>
<td>190.56</td>
<td>128</td>
<td>77</td>
</tr>
<tr>
<td>Simon-128/256</td>
<td>30/72</td>
<td>254.68</td>
<td>128</td>
<td>111</td>
</tr>
</tbody>
</table>
Perspectives

- Extend results to **Substitution Permutation Network** ciphers (AES, ...);
  
- Generalize the **State-test** technique;